



**Implementación de un Control de Posición para el Brazo Robótico Ufactory 850, a
través del SDK de Python**

Barreno Ubilluz, Xavier David y Landa Landa, Kevin Alexis

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Electrónica y Automatización

Trabajo de unidad de integración curricular, previo a la obtención del título de Ingeniero en
Electrónica y Automatización

Ing. Víctor Hugo Andaluz Ortiz PH.D.

29 de febrero del 2024

Latacunga



Plagiarism and AI Content Detection Report

Tesina_Barreno_Landa [1].pdf

Scan details

Scan time: March 1st, 2024 at 11:5 UTC
 Total Pages: 38
 Total Words: 9457

Plagiarism Detection

Types of plagiarism	Words
Identical	1.2% 117
Minor Changes	0.7% 71
Paraphrased	4.9% 463
Omitted Words	0% 0

AI Content Detection

Text coverage	Words
AI text	7.3% 697
Human text	92.7% 8800

[View details](#)

Plagiarism Results: (20)

[aa1_ifr_robots.pdf](#) 1%

https://www.ed.torres-srl.com.ar/sites/default/files/aa1_ifr_robots.pdf

Reporte especial | Robótica | Aplicación Estos son los robots de servicio Los robots de servicio están en alza. Durante los últimos años...

[UTA_af0badcdef77668e95132429d9f43d4c](#) 0.8%

https://maee.cedea.edu.ar/recond/uta_af0badcdef77668e95132429d9f43d4c

Skip to content My Title ...

[Robótica | Estos son los robots de servicio | Editores](#) 0.8%

https://www.ed.torres-srl.com.ar/servicios/aa1/ifr_robots_de_servicio

Skip to main content ...

[Andres-Ubeda_Teoria_RoboticaServicios.pdf](#) 0.8%

https://vva.ua.es/dspace/bitstream/10045/87662/1/andres-ubeda_teoria_roboticaservicios.pdf

AUBEDA

10/09/2018 Curso 2018/2019 ANDRES ÚBEDA CASTELLANOS Profesor Ayudante Doctor Grupo AURQVA

Template resources: Slides Carnival Content...

Certified by

About this report
help.copyleaks.com

copyleaks.com

Ing. Andaluz Ortiz, Víctor Hugo PH.D.

Director



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Electrónica y Automatización

Certificación

Certifico que el trabajo de unidad de integración curricular: **“Implementación de un Control de Posición para el Brazo Robótico Ufactory 850, a través del SDK de Python”** fue realizado por el señor **Barreno Ubilluz, Xavier David** y el señor **Landa Landa, Kevin Alexis**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 29 de febrero de 2024

Ing. Andaluz Ortiz, Victor Hugo PH.D.

C.C.: 1803737442



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Electrónica y Automatización

Responsabilidad de Autoría

Nosotros, **Barreno Ubilluz, Xavier David**, con cédula de ciudadanía n° 1723501639 y **Landa Landa, Kevin Alexis**, con cédula de ciudadanía n° 1804379384, declaramos que el contenido, ideas y criterios del trabajo de unidad de integración curricular: **“Implementación de un Control de Posición para el Brazo Robótico Ufactory 850, a través del SDK de Python”** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 29 de febrero del 2024

Barreno Ubilluz, Xavier David

C.C.: 1723501639

Landa Landa, Kevin Alexis

C.C.: 1804379384

Dedicatoria

El haber terminado esta etapa importante de mi vida se la dedico principalmente a Dios, fuente de sabiduría y guía constante en mi camino, a mi amada familia, cuyo amor y apoyo han sido mi fuerza inquebrantable para lograr el objetivo: a mi querida madre, Natali Ubilluz, por su sacrificio y dedicación incondicional; a mi valiente padre, David Barreno, por ser mi ejemplo de perseverancia, trabajo y determinación; a mis adoradas hermanas, Stefany, Katherin y Genesis, por compartir conmigo risas, lágrimas y sueños; y a mi abuelo paterno, Luis Barreno, cuya sabiduría, humildad y cariño han iluminado mi sendero, quien también actualmente es el único de todos mis abuelitos que Dios ha permitido que esté presente para celebrar este logro personal. A todos ustedes, desde el fondo de mi corazón, por brindarme la oportunidad de alcanzar este sueño tan anhelado, les dedico este logro importante en mi vida.

Xavier Barreno

Dedicatoria

"A mi amado Padre Celestial, primero Dios, quien me ha guiado y fortalecido a lo largo de este arduo pero gratificante camino académico.

A mi amado padre, Abelardo Landa, cuyo apoyo incondicional y sabios consejos han sido mi faro en los momentos de duda y dificultad. Su ejemplo de sacrificio, perseverancia y amor por el conocimiento ha sido mi inspiración constante.

A mi amada madre, Rosario Landa, cuyo amor incondicional, dedicación y sacrificio han sido el motor que impulsa mis sueños y aspiraciones. Sus palabras de aliento, su fuerza inquebrantable y su infinito amor han sido mi refugio en los momentos de desafío.

A ambos, Padre y Madre, les dedico este logro con todo mi corazón, su amor, guía y sacrificio han sido la base sobre la cual he construido mi camino para alcanzar esta meta.

Kevin Landa

Agradecimiento

En primer lugar, quiero expresar mi profundo agradecimiento a Dios, cuya guía y bendiciones han sido fundamentales en cada paso de este camino académico.

A mi amada familia, quienes han sido mi roca y mi inspiración a lo largo de esta travesía: a mi querida madre, Natali Ubilluz, por su amor incondicional y su constante apoyo; a mi padre, David Barreno, por ser mi ejemplo de trabajo, esfuerzo y dedicación; a mis adoradas hermanas, Stefany, Katherin y Genesis, por llenar mi vida de alegría y aliento en los momentos más difíciles.

A mi distinguido tutor de tesis, el PhD. Víctor Hugo Andaluz Ortiz, por su invaluable orientación, paciencia y confianza en mis capacidades. Su sabiduría y apoyo fueron fundamentales en el desarrollo de este trabajo.

A la PhD. Jessica Ortiz, esposa de mi tutor, agradezco sinceramente su ayuda, sus conocimientos y su amistad, que enriquecieron mi experiencia académica y personal.

Y, por último, pero no menos importante, quiero extender mi gratitud a mi compañero de tesis, Kevin Landa. Juntos, hemos enfrentado desafíos, celebrado logros y hemos llegado a este punto con determinación y perseverancia. Su colaboración y compañerismo han sido invaluable.

A todos ustedes, mi más sincero agradecimiento por formar parte de este viaje y por contribuir al éxito de este proyecto.

Xavier Barreno

Agradecimiento

En primer lugar, deseo expresar mi más profundo agradecimiento a Dios, por su amor incondicional, su constante guía y su inagotable provisión a lo largo de este viaje académico. Sin su bendición y dirección divina, este logro no sería posible.

A mi amado padre, Abelardo Landa, y a mi amada madre, Rosario Landa, les agradezco de todo corazón por su apoyo incondicional, sus palabras de aliento y su sacrificio infinito. Su amor y ejemplo han sido mi mayor inspiración y motivación para alcanzar mis metas.

A mi familia, por su inquebrantable apoyo y comprensión a lo largo de esta travesía. Cada uno de ustedes ha sido un pilar fundamental en mi vida, y su aliento constante ha sido mi impulso para seguir adelante, incluso en los momentos más difíciles.

A mi tutor, el Ing. Víctor Andaluz PH.D., por su orientación experta, su paciencia y su dedicación para guiarme a lo largo de este proceso de investigación. Sus valiosos consejos, su profundo conocimiento y su amistad han sido fundamentales para el desarrollo de esta tesis.

A la Ing. Jessica Ortiz PH.D., por su colaboración, sus ideas innovadoras y su apoyo enriquecedor durante el desarrollo de este trabajo. Su experiencia y perspectiva han contribuido significativamente a la calidad y relevancia de esta tesis.

A mis amigos que conocí durante toda mi carrera universitaria, especialmente a Jimmy, Jimson, Fernando, Andrés, Milton, Zeus, Alexis, Jenny, Jessy, Grace, Raphael quienes estuvieron a mi lado durante los momentos de estudio, celebración y desafíos. Su amistad y compañerismo han hecho de este viaje una experiencia inolvidable.

Por último, pero no menos importante, a mi compañero de tesis, Xavier Barreno, por su colaboración, trabajo en equipo y su sincera amistad han sido los pilares para la culminación de este proyecto académico, e igual forma gracias por todos los momentos de estudio, celebración y desafíos que hemos hecho durante toda la carrera.

A todos y cada uno de ustedes, les estoy eternamente agradecido. Su apoyo, amor y amistad han sido un regalo invaluable en mi vida y han hecho posible este logro. ¡Gracias de todo corazón!"

Kevin Landa

ÍNDICE DE CONTENIDO

Carátula.....	1
Reporte de verificación de contenido	2
Certificación	3
Responsabilidad de Auditoria.....	4
Autorización de Publicación	5
Dedicatoria.....	6
Dedicatoria.....	7
Agradecimiento	8
Agradecimiento	9
Índice de Contenido	11
Índice de Figuras	14
Índice de Tablas	16
Resumen	17
Abstract.....	18
Capítulo I: Introducción	19
Antecedentes.....	20
Planteamiento del Problema y Justificación	24
Hipótesis	24
Objetivos	25
<i>Objetivo General</i>	25
<i>Objetivos Específicos</i>	25
Capítulo II: Marco Teórico	26

Robot Manipulador	28
Morfología de un robot.	28
<i>Articulaciones</i>	28
<i>Tipos Comunes de Articulaciones</i>	28
<i>Rotacional o de Revolución</i>	28
<i>Grados de libertad</i>	30
<i>Grados de libertad articulares</i>	30
<i>Grados de libertad espaciales</i>	31
Actuadores	31
<i>Actuadores Eléctricos</i>	31
<i>Los motores AC Industriales</i>	32
<i>Los motores de corriente continua (DC)</i>	32
<i>Actuadores Hidráulicos</i>	33
<i>Actuadores Neumáticos</i>	34
Sensores	34
<i>Sensores propioceptivos</i>	34
<i>Sensores exteroceptivos</i>	35
<i>Sensores de fuerza y torque</i>	35
<i>El sensor de colisión</i>	35
<i>Sensores de visión</i>	35
Elementos de control final.....	36
Herramientas de los robots para diferentes tareas.....	37
<i>Pistola para pintar superficies</i>	37

<i>Soldadura</i>	37
<i>Mecanizado</i>	38
<i>Cañón de agua a presión</i>	39
El espacio de trabajo	39
Robots de servicio	40
Capítulo III: Desarrollo	42
Instalación del software e integración de los SDK Python para el robot Ufactory 850 55	
Capítulo IV: Pruebas y Resultados	65
Capítulo V: Conclusiones y Recomendaciones	72
Conclusiones	72
Recomendaciones	73
Bibliografía	74
Anexos	76

ÍNDICE DE FIGURAS

Figura 1 <i>Robot Industrial Kuka</i>	27
Figura 2 <i>Robot colaborativo ensamblado ah un robot móvil</i>	27
Figura 3 <i>Tipos de articulaciones de un robot.</i>	29
Figura 4 <i>Grados de libertad articulares</i>	30
Figura 5 <i>Motores eléctricos en robots manipuladores.</i>	31
Figura 6 <i>Motor AC de un Robot Kuka</i>	32
Figura 7 <i>Motores eléctricos en robots manipuladores, dynamixel</i>	33
Figura 8 <i>Actuador hidráulico HKS</i>	33
Figura 9 <i>Actuador neumático lineal Festo DFPC</i>	34
Figura 10 <i>Sensores propioceptivos encoder</i>	35
Figura 11 <i>Sensores exteroceptivos</i>	36
Figura 12 <i>Elementos de control final</i>	36
Figura 13 <i>Herramienta pistola pintura en robótica industrial</i>	37
Figura 14 <i>Herramienta soldadora en robótica industrial</i>	38
Figura 15 <i>Herramienta fresadora en robótica industrial</i>	38
Figura 16 <i>Herramienta corte por chorro en robótica industrial</i>	39
Figura 17 <i>Espacio de trabajo de un robot antropomórfico</i>	40
Figura 18 <i>Robot de servicio domésticos.</i>	40
Figura 19 <i>Robots de cirugía en hospitales</i>	41
Figura 20 <i>Movimientos en X/ Y / Z de Roll, Pitch y Yaw con sistema de referencia desde la base</i>	42
Figura 21 <i>Ejemplo gráfico de la rotación del robot en Roll, Pitch y Yaw.</i>	43
Figura 22 <i>Arquitectura de la conexión externa del robot para el funcionamiento</i>	47
Figura 23 <i>Caja de control del Robot UFactory 850</i>	48
Figura 24 <i>Diagrama de pines de la caja de control</i>	50
Figura 25 <i>Espacio de trabajo de un robot UFactory 850</i>	51
Figura 26 <i>Dimensiones del Robot UFactory 850</i>	52

Resumen

Este trabajo se enfocó en una exploración detallada del Software Development Kit (SDK) de Python para el brazo robótico Ufactory 850, con el objetivo principal de implementar y ejecutar tareas específicas de posicionamiento mediante esta herramienta. En primer lugar, se llevó a cabo una investigación rigurosa para comprender completamente las capacidades y características del SDK de Python, así como para examinar el funcionamiento y las especificaciones técnicas del brazo robótico Ufactory 850. Posteriormente, se procedió a la implementación de una tarea concreta de posicionamiento, lo que implicará el diseño y la ejecución de secuencias de comandos destinadas a controlar los movimientos del brazo robótico y a posicionar objetos en ubicaciones predefinidas de manera precisa. Además, se realizó una evaluación experimental exhaustiva de la tarea de posicionamiento, donde se analizarán en profundidad la precisión, la eficiencia y la confiabilidad del proceso de posicionamiento utilizando el SDK de Python en cuestión. Los resultados obtenidos permitirán una comprensión detallada de la funcionalidad y el rendimiento del SDK de Python en relación con el brazo robótico Ufactory 850, lo que a su vez facilitará la identificación de posibles áreas de mejora u optimización para su aplicación en entornos industriales y servirá como punto de partida para investigación e implementación de trabajos futuros.

Palabras Clave: SDK de Python, brazo robótico, Ufactory 850.

Abstract

This work focused on a detailed exploration of the Python Software Development Kit (SDK) for the Ufactory 850 robotic arm, with the primary goal of implementing and executing specific positioning tasks using this tool. First, a rigorous investigation was conducted to fully understand the capabilities and features of the Python SDK, as well as to examine the operation and technical specifications of the Ufactory 850 robotic arm. Subsequently, the implementation of a concrete positioning task was undertaken, which will involve the design and execution of scripts aimed at controlling the movements of the robotic arm and positioning objects in predefined locations in a precise manner. In addition, a thorough experimental evaluation of the positioning task was performed, where the accuracy, efficiency and reliability of the positioning process will be analyzed in depth using the Python SDK in question. The results obtained will allow a detailed understanding of the functionality and performance of the Python SDK in relation to the Ufactory 850 robotic arm, which in turn will facilitate the identification of possible areas of improvement or optimization for application in industrial environments and serve as a starting point for research and implementation of future work.

Keywords: Python SDK, robotic arm, Ufactory 850,

Capítulo I

Introducción

A partir de la industria 3.0 el protagonismo del robot en el área industrial fue un eje principal de investigación, puesto que las ventajas en trabajos en cadena fueron reconocidos a nivel mundial, desde ese punto la evolución de la robótica no se ha detenido y ha permitido que los trabajos robóticos puedan ser realizados en áreas diferentes a la industria, dando paso a una revolución tecnológica como lo es la robótica de servicio. Los prototipos robóticos que se desarrollan en el área de servicio se basan en desarrollar tareas que ayudan y facilitan a los clientes en actividades repetitivas que son consideradas de alta peligrosidad; definidas también como complejas para ser echas por personas. Uno de los prototipos principales de esta familia son los brazos robóticos, debido a que presentan características similares a las extremidades superiores de un humano con la diferencia de que estos modelos pueden trabajar por largos tiempos sin sufrir desgaste físico ni moral. Por tal motivo existen multinacionales que fabrican robots manipuladores que diseñan prototipos industriales pero que se pueden implementar a trabajos menos complejos dando más facilidad económica a los clientes para que puedan ser ejecutados en varios avientes industriales.

En la actualidad la robótica y la automatización, los robots manipuladores ejercen tareas importantes que fundamentan un avance en la efectividad y la capacidad de producción en las industrias. El brazo robótico UFactory 850, con su diseño modular y versatilidad, se presenta como una opción atractiva para aplicaciones en el ámbito de la robótica de servicio por el motivo que el robot no presenta una robustes que le permita ser empleado en la industria. La investigación de esta tesina se basa en la indagación y ejecución de un sistema de control para el robot UFactory 850 utilizando los Kits SDK de Python. El dispositivo será programado para una tarea de paletizado, donde ubique de manera ordenada y secuencial, productos de diferente tamaño en casilleros según corresponda.

El proyecto se llevará a cabo utilizando Python, un lenguaje de programación interpretado y compatible con múltiples plataformas. Visual Studio Code ofrece un entorno integrado para Python, lo que elimina la necesidad de instalar emuladores o adaptadores adicionales para ejecutar scripts de Python. Esto es beneficioso para desarrollar algoritmos de control abiertos para el movimiento de brazos robóticos en software de código abierto.

Antecedentes

La robótica tiene fundamentos culturales arraigados, a lo largo de los siglos, se ha observado un esfuerzo constante por parte de la humanidad para encontrar sustitutos capaces de imitar su comportamiento en diversas interacciones con el entorno circundante (Siciliano, 2010). Distintos motivos han impulsado esta búsqueda continua, haciendo referencia a principios filosóficos, económicos, sociales y científicos; como resultado, el término "robot" hizo su primera aparición en la obra de teatro checa de ciencia ficción de 1920 y fue el principio de una idea y apertura clara hacia la robótica.

Cuando se reflexiona sobre los robots, la imagen común es la de grandes y potentes máquinas que martillean piezas de metal o instalan puertas de automóviles, etc. Los robots han desempeñado un papel fundamental en la fabricación durante más de 50 años, aplicándose a tareas específicas con el objetivo de aumentar la productividad. (IFR, International Federation of Robotics, 2021) No obstante, en la última década, se ha experimentado una transformación significativa en las capacidades y aplicaciones de estos dispositivos.

Los prototipos de robots manipuladores a medida que pasa el tiempo, más ocupan puestos de operadores de producción, en los que se desempeñan los seres humanos y esto se basa por el principio de que su estructura dimensional se asemeja bastante a la de un brazo humano, es decir, una extremidad superior, pero con la diferencia que se componen de elementos industriales capaces de trabajar indefinidamente sin cansancio y sin remuneración. Es por eso que los fabricantes de estos robots manipuladores los fabrican con varias dimensiones que varían desde pequeños hasta 10 veces más grandes y con la

capacidad de operar en áreas industriales desde la biotecnología hasta el tratamiento de piezas industriales. Estos pueden ser operados por elementos que implementan pequeños o micro robots (Matjaž Mihelj, 2010) que permiten a los operarios realizar tareas de contracción, succión, posición entre otros, creando así dos grandes grupos de estudio como lo es la robótica de servicio y la robótica industrial.

La robótica de servicio se ha impuesto para los usuarios que consumen la robótica, para generar beneficios como una ventana de investigación con el fin de dar interacción a placer de los seres humanos en diversos espacios. A diferencia de los autómatas que se desempeñan en el ámbito industrial que trabajan en espacios delimitados y especializados, los prototipos en el área de servicio están asignados a trabajar en espacios más complejos y ayudar de forma directa a los humanos en las tareas cotidianas. (Alvarez, 2023) Estos prototipos se posicionaron valiosamente en varios entornos, desde la prestación de servicios en el área de la medicina y ayuda domésticas hasta en el ámbito de hotelería y la ayuda en la inteligencia militar, teniendo así un gran potencial que mejora la calidad de vida de las personas ahorrando procesos en la sociedad. En la actualidad las iniciativas tecnológicas de la robótica de servicio, a más de hacer automáticos los procesos busca incrementar la eficacia de la seguridad con la intervención humana, dejando una huella importante a la hora de experimentar tecnología en las tareas cotidianas de los seres humanos.

Ahora, la otra rama de estudio que es la robótica industrial se desempeña únicamente en industrias de desarrollo en masa por el motivo que brinda aportes tecnológicos normalizados propios de procesos de automatización industrial. (Andrius Dzedzickis, Ernestas Šutinys, Urte Samukaite-Bubniene and Vytautas Bucinskas, 2022) Se describe como un prototipo robótico industrial a un manipulador reprogramable y controlable de manera automática, que tiene la capacidad de programación en varios puntos, y con la posibilidad de ser fijo o móvil para su ejecución en varias aplicaciones de automatización industrial según el estándar de la norma ISO 8373 actualizada en el año 2012.

La robótica industrial ha superado las expectativas de los usuarios, superando límites de producción industrial, donde los dueños de las fábricas están satisfechos con la producción y ganancias que le dejan la automatización por medio de la robótica.

Actualmente los prototipos de robots industriales además de representar elementos propios en los KPI's de producción, también trabajan como medios que transforman e impulsan la exactitud, la eficacia y forma versátil en distintos lugares. (Siciliano, 2010) Los prototipos robóticos industriales avanzados, detallados por su versatilidad de ejecutar tareas con cierto grado de complejidad, están iniciando grandes cambios notables en la manera en que se adquiere y ejecutan procesos altamente productivos. (Nayely Morales-Ramirez, 2023)

En el proceso de investigación entre la relación directa que tiene la ingeniería, la inteligencia artificial y la automatización de procesos, la robótica industrial aparte de mejorar la producción, define las fronteras de lo posible en los sistemas de manufactura y producción. Entonces a partir de lo establecido anteriormente en el mundo se generó una cuarta revolución industrial, misma que generó una conmoción en la industria de la robótica adjuntando a su trabajo a la inteligencia artificial y al internet de las cosas originando la industria 4.0.

La industria 4.0 por medio de sus aplicaciones y tecnología dan la posibilidad de comunicación en redes interconectadas por ordenadores, equipos de control electrónico, aplicaciones con algoritmos de control y los seres humanos, esto posibilita a los usuarios una visión más clara de los procesos críticos avanzados dando paso a varias soluciones. (Anand Nayyar, Akshi Kumar, 2020) Los operadores tendrán los datos informativos necesarios en el momento que corresponda con la finalidad de evitar inconvenientes como demoras y reduciendo significativamente los tiempos de procesamiento. La innovación de la robótica en la tecnología que la maneja se ejecuta para emplear varias tareas en las industrias que la incorporan. Al entender de manera activa los problemas que se generan en la fabricación y comprender los detalles de producción entre el personal de trabajo, se puede realizar muchas mejoras continuas para la reducción de costos y aumento de la

producción. Generar una visión integral de las acciones de la empresa será tomado como una ayuda a repotenciar la competitividad y a favorecer un aumento considerable de la eficacia. Si bien en la industria 4.0, cada artefacto, equipo electrónico y ordenadores está directamente unido a una estrategia de producción industrial que forma un lugar apropiado para la recolección de datos y procesamiento de los mismos a gran escala. Las fábricas por medio de su grupo gerencial indagan los límites de hasta dónde puede llegar hoy en día la robótica y anexarla a la industria 4.0.

La cuarta revolución industrial con la industria 4.0 cambio la forma de trabajo de las fábricas en todo el mundo, por medio de todos los cambios que se generaron, se da inicio al internet global de las cosas (Iot). Esta aplicación tiene la capacidad de guardar, deducir y conducir un gran número de datos recopilados en el proceso de producción de manera detallada, transformando las operaciones en procesos eficientes, precisos y óptimos. Pero algunos de los retos más extremos de la industria pueden asimilarse muy poco prácticas, en la actualidad hay un sin número de soluciones para ejecutar las aplicaciones que brinda la industria 4.0, generalmente en el ámbito de la robótica industrial, por medio de procesos previamente establecidos. Las empresas que fabrican a los autómatas también han creado opciones para la mejora óptima de la industria 4.0 como una base digital donde se brinda asistencia a los usuarios para agregar y ejecutar los datos de manera segura mezclando conjuntamente con datos industriales más grandes, incorporando análisis de base de datos y generando información para mejorar de manera óptima el rendimiento y la productividad. (Andrius Dzedzickis, Ernestas Šutinys, Urte Samukaite-Bubniene and Vytautas Bucinskas, 2022)

Ante todos los beneficios que brinda la robótica en la industria 4.0 se puede afirmar que existe consecuencias notables en el ámbito social, el costo tiende a ser más significativo para el empleo de personas. Además, libera personal en otros campos industriales, como la fabricación, programación y mantenimiento, para aplicar sus

habilidades y experiencia. Los robots industriales se integran generalmente en diversas actividades.

Planteamiento del Problema y Justificación

La implementación de algoritmos con SDK de Python da la posibilidad de poder implementar varios algoritmos de control para la manipulación del Robot UFactory 850 y no limitarse a los movimientos que entrega el fabricante con el software propietario, esto permite que se pueda realizar tareas más complejas con algoritmos de estudio que abra la posibilidad a los usuarios de comprobar los diferentes métodos de estudio.

En el ámbito de control de procesos es necesario el desarrollo de brazos robóticos con la capacidad de operar de manera eficiente en entornos no estructurados donde se requiere soluciones flexibles que permitan a los robots de servicio desempeñen tareas de un robot industrial, donde la disposición del entorno y los objetos puede cambiar a otros ambientes no industriales. Lo que se plantea es la necesidad de que el robot de servicio pueda adaptarse y cumplir con funciones asignadas de manera precisa.

La flexibilidad inherente a Python permitirá la implementación de algoritmos de control que son esenciales para la operación exitosa del robot Ufactory 850 en entornos no estructurados. Además, la integración de técnicas de aprendizaje por computadora a través de bibliotecas de Python especializadas potenciará la capacidad del robot para adaptarse a situaciones y realizar tareas de manera autónoma.

Hipótesis

La implementación de un esquema de control de posición a través del SDK de Python. permite ejecutar tareas autónomas de paletizado a través del brazo robótico Ufactory 850.

Objetivos

Objetivo General

- Implementar una tarea de posición para el brazo robótico Ufactory 850, a través del SDK de Python.

Objetivos Específicos

- Investigar la funcionalidad del SDK de Python.
- Implementar un algoritmo, a fin de ejecutar tareas de posicionamiento del brazo robótico Ufactory 850.
- Evaluar experimentalmente la tarea de posicionamiento implementado a través del SDK de Python.

Capítulo II

Marco Teórico

El Instituto de Robótica definió el robot en 1971 como “Un robot manipulador es reprogramable y multifuncional diseñado para desplazar materiales, partes, herramientas o dispositivos especializados a través de varios movimientos programados para el desarrollo de una variedad de tareas para realizar operaciones que antes sólo podían realizar los seres humanos”. (Americas, 2020)

Los primeros robots se remontan desde el siglo III a.C., considerando que las primeras máquinas están más relacionadas con la parte mecánica que con la robótica. (Claut, 2022), robot humanoide se llamó Elektro fue construido por la Westinghouse. Era una máquina de aproximadamente dos metros que podía caminar y grababa unas 700 palabras y simulaba una conversación con el usuario. (IFR, International Federation of Robotics, 2021)

Los robots fijos industriales que hacen tareas predeterminadas propuestas por el desarrollador, según se muestra en la Figura 1, están presentes en toda el área de fábricas en la parte de automatización y están establecidas para funcionar en zonas estructuradas para todo tipo de proceso, los robots de servicio van incrementándose y se hace más novedoso y ciertos brazos robóticos colaborativos realizan su función en un entorno no estructurado y están ensamblados a robots móviles ver Figura 2

Robot Manipulador

El robot manipulador se utiliza para diferentes aplicaciones incluso con acceso limitado y se caracterizan por tener eslabones que son articulados y esto se compara al de un brazo humano, otra característica es que puede moverse de forma automática a través de un sistema de control y elementos terminales (Claut, 2022)

Morfología de un robot.

Al mencionar la morfología de un robot se debe considerar que un robot está formado por los siguientes elementos: estructura, actuadores, sensores, grados de libertad, espacio de trabajo entre otros. (Sino-Inst, 2022)

Articulaciones

Las articulaciones son conexiones entre eslabones, que permiten la movilidad entre ellos. las articulaciones desempeñan una función del movimiento del robot manipulado, (Alvarez, 2023)

Tipos Comunes de Articulaciones:

Prismática o Lineal:

Permite que la articulación tenga un movimiento de traslación lineal a lo largo de un eje.

Rotacional o de Revolución:

Permite el giro o rotación alrededor de un eje.

Tornillo sin fin

La articulación de tornillo sin fin en robots se refiere a la implementación de un mecanismo de transmisión basado en un tornillo sin fin para lograr movimiento rotativo en una junta o articulación del robot. (Matjaž Mihelj, 2010)

Articulación Planar:

Restringe el movimiento a un plano.

Grados de libertad

Se refiere al número de variables de posición independiente a especificarse para localizar todas las piezas y se definen como los posibles movimientos básicos como giros y desplazamientos.

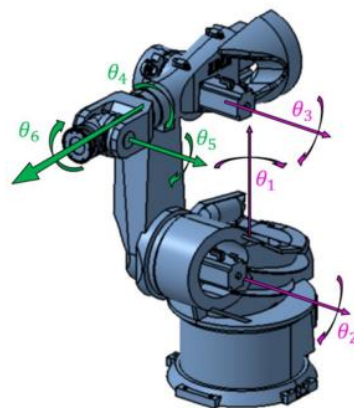
Grados de libertad articulares

Los grados de libertad articulares se refieren a la cantidad de movimientos independientes que realiza el robot en cada una de sus articulaciones. Cada articulación proporciona una dirección de movimiento o rotación. (Olier Caparroso Iván, 2021)

En un robot cada articulación como el hombro, el codo y la muñeca agrega un grado de libertad. Esto significa que el robot puede moverse tridimensionalmente independientemente en su espacio de trabajo como se muestra en la Figura 4. (Alvarez, 2023)

Figura 4

Grados de libertad articulares



Nota. Tomado de (Nayely Morales-Ramirez, 2023)

Actuadores Neumáticos

Los actuadores neumáticos como se muestra en la Figura 9 usan aire comprimido para realizar el movimiento. Son empleados en robótica donde se realiza actividades más ligeras como el traslado de objetos.

Figura 9

Actuador neumático lineal Festo DFPC.



Nota. Tomado de (INESEM, 2024)

Sensores

Los sensores son instrumentos que se encuentran en sistemas de control y son sistemas de percepción que brindan información. Se pueden encontrar desde sensores para detectar objetos, hasta sensores de visión artificial, estos sensores se clasifican en dependiendo de la zona que se encuentra del robot.

Sensores propioceptivos

Se refiere a cualquier sensor usado para medir internamente al robot. Monitoreando esta clase de sensores puede indicar al robot las velocidades que el robot tiene en las articulaciones como los encoders dan información de la posición y orientación que se muestra en la Figura 10

Figura 10

Sensores propioceptivos encoder



Nota. Tomado de (Olier Caparros Iván, 2021)

Sensores exteroceptivos

Estos sensores permiten a los robots sentir el entorno que les rodea. Pueden detectar variables como la posición, la velocidad, temperatura, entre otras.

Se pueden encontrar sensores como:

Sensores de fuerza y torque:

Permiten medir fuerzas del robot y los momentos que actúan sobre un objeto. Estos sensores son importantes para la manipulación de objetos que se requiera saber cuánto de fuerza se está aplicando al mismo.

El sensor de colisión

Se lo ubica en el extremo operativo del robot y tiene como objetivo minimizar lo más posible el daño de la garra, ventosa o gripper.

Sensores de visión

Son sistemas de visión que permiten al robot visualizar el entorno que le rodea. Estos sensores son utilizados para la realización de tareas de inspección o detección de objetos.

Figura 11*Sensores exteroceptivos*

a) Sensor de fuerza

b) Sensor de colisión

c) Cámara

Nota. Tomado de (Matjaž Mihelj, 2010)**Elementos de control final**

Los elementos de control final de un robot manipulador en robótica constituyen una parte esencial del sistema, desempeñando un papel crucial en la ejecución precisa y eficiente de tareas específicas. Estos elementos están diseñados para gestionar y supervisar el comportamiento del efector final del robot, que es la parte del manipulador responsable de interactuar con el entorno y realizar acciones específicas, como ventosas, gripper o garras, como los que se muestra en la Figura 12 (Americas, 2020)

Figura 12*Elementos de control final*

a) Ventosa

b) Gripper

c) Garra

Nota. Tomado de (Jazar, 2021)

Herramientas de los robots para diferentes tareas.

En la industria los robots se dedican a la aplicación de mecanizados mediante herramientas para lo cual la robótica industrial tiene las siguientes herramientas.

Pistola para pintar superficies.

Una de las funciones en robótica industrial es el uso de robots para aplicar pinturas en piezas de diseño complejo y de gran magnitud.

La robótica industrial permite realizar una secuencia de movimientos adecuada para que la pistola pueda aplicar la pintura de forma adecuada como se muestra en la Figura 13

Figura 13

Herramienta pistola pintura en robótica industrial



Nota. Tomado de (INESEM, 2024)

Soldadura

La herramienta de soldadura es utilizada para soldar elementos metálicos y son dos las principales ventajas del uso de robots con herramientas de soldadura, la primera es la gran maniobrabilidad y la segunda el trabajo en un ambiente peligroso a altas temperaturas.

Figura 14

Herramienta soldadora en robótica industrial



Nota. Tomado de (INESEM, 2024)

Mecanizado

Da la posibilidad de incluir estas herramientas abre el campo de fabricación automatizada a cualquier tipo de pieza por compleja que sea su estructura y morfología como se muestra en la Figura 15. De modo que podemos incrementar la producción de piezas más complejas.

Figura 15

Herramienta fresadora en robótica industrial



Nota. Tomado de (INESEM, 2024)

Un robot de servicio para uso profesional es un robot utilizado para tareas comerciales, generalmente operado por un operador entrenado. Los ejemplos incluyen robots que realizan limpieza para lugares públicos, en oficinas u hospitales también se considera robots bomberos, robots de rehabilitación o robots de cirugía en los hospitales como se muestra en la Figura 19. (Anand Nayyar, Akshi Kumar, 2020)

Figura 19

Robots de cirugía en hospitales



Nota. Tomado de (Claut, 2022)

Capítulo III

Desarrollo

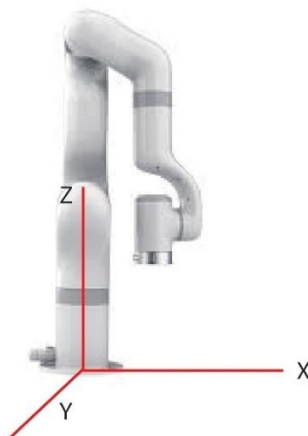
Morfología del Robot UFactory 850.

La morfología del robot UFactory 850 representa un avance destacado en la ingeniería robótica, fusionando diseño innovador con funcionalidad de vanguardia. Este robot se distingue por su estructura modular y versátil, diseñada para adaptarse a una amplia gama de aplicaciones de la robótica de servicio y de investigación. Empezando desde las articulaciones hasta los sensores del robot optimizan la eficiencia y la flexibilidad en entornos dinámicos.

El funcionamiento del robot se basa en tres movimientos específicos como Roll, Pitch y Yaw, mismos que rotan secuencialmente alrededor de las $X / Y / Z$ del sistema de coordenadas desde la base del robot tomando como referencia la Figura 20

Figura 20

Movimientos en $X / Y / Z$ de Roll, Pitch y Yaw con sistema de referencia desde la base.



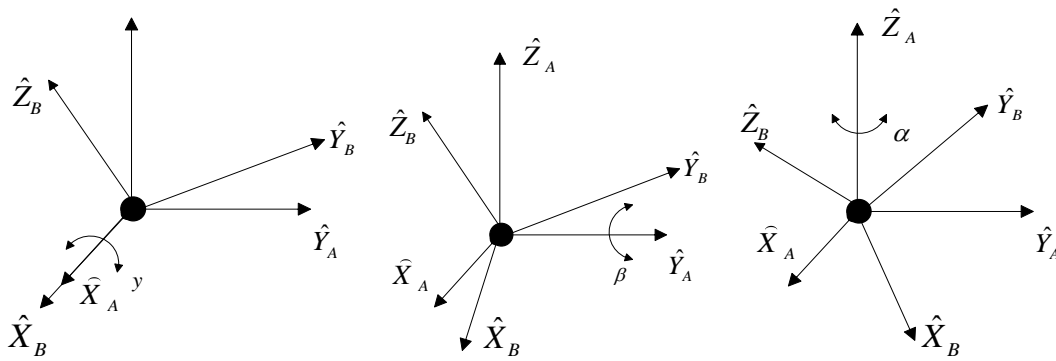
Nota. Tomado de (Ufactory, 2022)

La Figura 21 es la representación de la orientación de Roll, Pitch y Yaw de $\{B\}$ respecto a $\{A\}$ viene dado por el siguiente ejemplo, primero se superponen el sistema de

coordenadas $\{B\}$ y un sistema de coordenadas de referencia conocido $\{A\}$. Primero se gira $\{B\}$ alrededor de \hat{X}_A en y , luego alrededor de \hat{Y}_A en β , y finalmente alrededor de \hat{Z}_A en α . Cada rotación es alrededor de un eje fijo del sistema de coordenadas de referencia $\{A\}$.

Figura 21

Ejemplo gráfico de la rotación del robot en Roll, Pitch y Yaw.



Nota. Tomado de (Ufactory, 2022)

Del proceso representado en el ejemplo anterior se obtiene la matriz de rotación equivalente:

$${}^A_B R_{XYZ}(y, \beta, \alpha) = R_Z(\alpha) R_Y(\beta) R_X(y)$$

Nota: y corresponde a roll; β corresponde a pitch; α corresponde a yaw.

R_x , R_y y R_z son tres valores que se utilizan para representar la posición (pero no tres ángulos de rotación), que es el producto de un vector de rotación tridimensional $[x, y, z]$ y un ángulo de rotación $[\varphi(\text{escalar})]$. El ángulo del eje cumple características intrínsecas, se supone que el eje de rotación es $[x, y, z]$, y el ángulo de rotación es φ , entonces la orientación del ángulo axial se representa de la siguiente manera:

$$[R_x, R_y, R_z] = [x\varphi, y\varphi, z\varphi].$$

donde:

1. $[x, y, z]$ es un vector unitario, y φ es un valor no negativo.
2. La longitud del vector (módulo) de $[R_x, R_y, R_z]$ puede utilizarse para estimar el ángulo de rotación, y la dirección del vector es la dirección de rotación.
3. Si desea expresar la rotación inversa, invierta el vector del eje de rotación $[x, y, z]$, y el valor de φ permanecer inalterado.
4. Usando φ y $[R_x, R_y, R_z]$ también se puede derivar la representación de la actitud como cuatemo unitario

$$q = \left[\cos\left(\frac{\varphi}{2}\right), \sin\left(\frac{\varphi}{2}\right) * x, \sin\left(\frac{\varphi}{2}\right) * y, \sin\left(\frac{\varphi}{2}\right) * z \right]$$

Una vez establecido los puntos clave de movimientos y rotación mediante ecuaciones asociados con el robot UFactory 850, es fundamental comprender los parámetros de movimiento. Estos parámetros, que delinean la capacidad y versatilidad del robot, son esenciales para la programación y control preciso de sus movimientos en el entorno operativo. Desde la velocidad y aceleración hasta la configuración de trayectorias y la gestión de coordenadas, se detalla cómo estos elementos determinan la dinámica y eficiencia del robot UFactory 850 en la ejecución de tareas específicas.

Tabla 1

Rango de trabajo de cada articulación del brazo robótico UFactory 850.

	Articulaciones	UFactory 850
Máximo movimiento general		180° / s
Rango de Trabajo	Articulación 1	±360°
	Articulación 2	-132° ~ 132°
	Articulación 3	-242° ~ 3,5°
	Articulación 4	±360°
	Articulación 5	-124° ~ 124°
	Articulación 6	±360°

Tabla 2

Rango de diversos parámetros de movimiento del brazo robótico UFactory 850.

	Movimiento TCP	Movimiento Conjunto
Velocidad	0 ~ 1000mm / s	0 ~ 180° / s
Aceleración	0 ~ 50000mm / s ²	0 ~ 1145° / s ²
Sacudida o Tirón	0 ~ 100000mm / s ²	0 ~ 28647° / s ³

De los parámetros del brazo robótico UFactory 850, se destacan dos observaciones clave. En primer lugar, los comandos de movimiento TCP deben ser dirigidos por un SDK, si un comando de movimiento implica tanto la transformación de la posición como la

transformación de la actitud, la velocidad de rotación de la actitud es generalmente calculada automáticamente por el sistema. En este caso, el parámetro de velocidad especificado es la velocidad lineal máxima, rango de: $0 \sim 1000\text{mm} / \text{s}$.

En segundo lugar, cuando el movimiento TCP previsto sólo cambia la actitud (roll, pitch, yaw), con la posición (x, y, z) se mantiene sin cambios, la velocidad específica es la velocidad de rotación de actitud, por lo que el rango de 0 a 1000 corresponder con un movimiento de 0 a $180^\circ / \text{s}$.

La eficiencia y precisión del brazo robótico UFactory 850 se ven sustentadas por la definición de sus parámetros, cada uno expresado con unidades estándar de medida que facilitan la comprensión y programación del robot. En este contexto, se presenta la siguiente tabla de unidades predeterminadas en Python para cada uno de los parámetros del robot y su protocolo de comunicación.

Tabla 3

Parámetros en Python y protocolo de comunicación del robot Ufactory 850

Parámetros	Python - SDK	Protocolo de comunicación
$X / Y / Z$	Milímetros (mm)	Milímetros (mm)
Roll / Pitch / Yaw	Grados ($^\circ$)	Radianes (rad)
$J_1 / J_2 / J_3 / J_4 / J_5 / J_6 / J_7$	Grados ($^\circ$)	Radianes (rad)
Velocidad TCP	mm / s	mm / s
Aceleración TCP	mm / s^2	mm / s^2

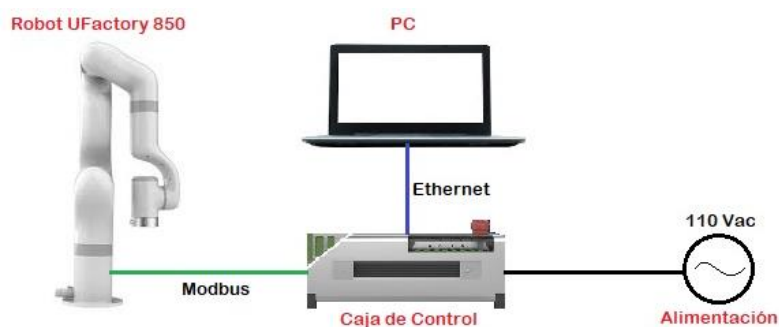
Parámetros	Python - SDK	Protocolo de comunicación
ASacudida o Tirón TCP	mm/s^3	mm/s^3
Velocidad Conjunta	$°/s$	rad/s
Aceleración Conjunta	$°/s^2$	rad/s^2
Sacudida o Tirón Conjunta	$°/s^3$	rad/s^3

Conocidos los detalles de los parámetros fundamentales que guían el rendimiento dinámico del robot UFactory 850, es importante dar atención a los componentes de hardware que constituyen la esencia de su funcionamiento. Cada componente fue diseñado para funcionar con los parámetros anteriormente descritos, creando una plataforma robótica avanzada y altamente eficiente.

Cada componente del hardware cumple un papel esencial en la ejecución coherente y precisa de las tareas asignadas al robot UFactory 850. Se presenta la arquitectura externa del robot en la Figura 22, destacando la sinergia entre los componentes para ofrecer un rendimiento óptimo en una variedad de aplicaciones industriales y de investigación.

Figura 22

Arquitectura de la conexión externa del robot para el funcionamiento.

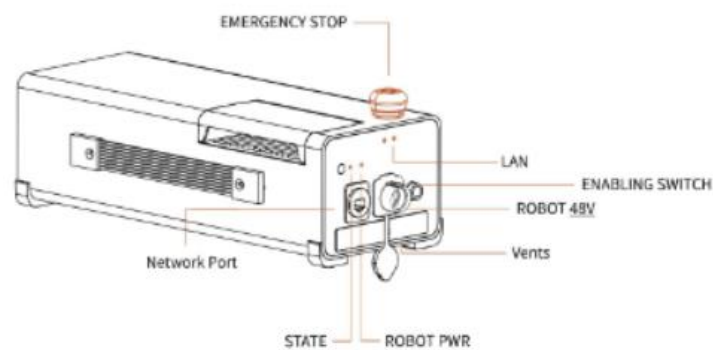


Nota. Tomado de (Ufactory, 2022)

La caja de control del robot UFactory 850 desempeña un papel central en su funcionamiento, siendo el núcleo que dirige las diferentes funciones del prototipo. Cada componente dentro de esta caja cumple una función distintiva para el rendimiento general del robot. Desde el procesamiento de datos hasta la gestión de la energía y la coordinación de los movimientos, la caja de control se posiciona como el cerebro del sistema, asegurando una ejecución precisa y eficiente de las tareas asignadas al UFactory 850 ver Figura 23 y 24.

Figura 23

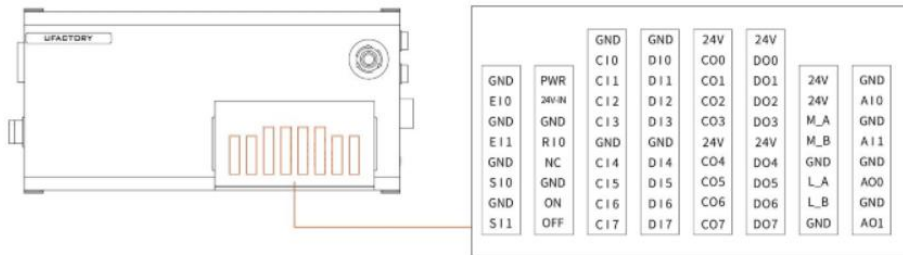
Caja de control del Robot UFactory 850



Nota. Tomado de (Ufactory, 2022)

Figura 24

Diagrama de pines de la caja de control



Nota. Tomado de (Ufactory, 2022)

Para la instalación del robot se debe tomar en cuenta las siguientes observaciones que influyen de manera directa en el funcionamiento y conexión de los equipos que intervienen en la arquitectura antes mencionada.

- ✓ Definir un espacio de trabajo para el prototipo.
- ✓ Fijar la base del brazo robótico.
- ✓ Conectar el brazo robótico con la caja de control.
- ✓ Conectar la caja de control con el cable de poder.
- ✓ Instalar el efector final

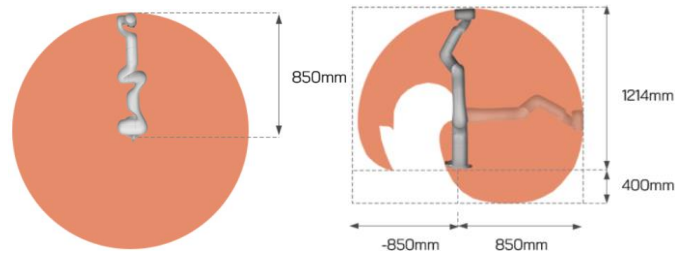
La definición de un espacio físico de trabajo para el robot UFactory 850 es un paso crítico que establece las bases para su desempeño eficiente y seguro. Este espacio, cuidadosamente delimitado, no solo proporciona los límites operativos del robot, sino que también garantiza la integridad de las operaciones y la seguridad del entorno circundante. La precisión en la definición de este espacio no solo optimiza la utilización del robot, sino que también reduce el riesgo de colisiones y maximiza la eficiencia en tareas específicas.

El área de operación del brazo robótico se define como la región abarcada por la extensión de sus eslabones. La Figura 25 ilustra las dimensiones y el alcance de trabajo del brazo. Al llevar a cabo la instalación, es esencial considerar cuidadosamente el rango de

movimiento del brazo para evitar posibles colisiones con personas y equipos circundantes (se excluye el extremo efectivo del alcance operativo).

Figura 25

Espacio de trabajo de un robot UFactory 850



Nota. Tomado de (Ufactory, 2022)

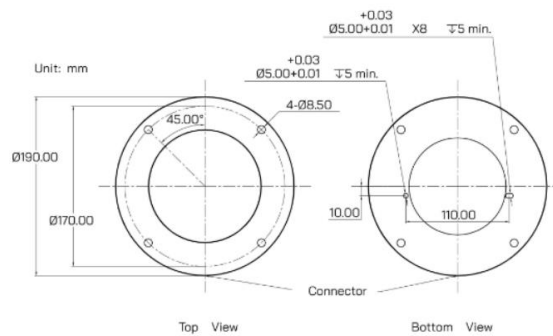
El espacio de trabajo del Robot está definido por 850 mm, es por ese motivo que el robot lleva el nombre de UFactory 850.

La instalación de la base del robot UFactory 850 es un paso esencial para establecer una plataforma robusta y segura. Este proceso, fundamentado en la disposición precisa y la fijación estable, sienta las bases para el rendimiento óptimo del robot en su entorno de trabajo.

El brazo robótico tiene cuatro pernos M8 suministrados y se puede montar a través de cuatro orificios $\varnothing 8,5$ en la base del brazo robótico. Se recomienda apretar estos pernos con un par de apriete de 20N-m.ver Figura 26

Figura 26

Dimensiones del Robot UFactory 850

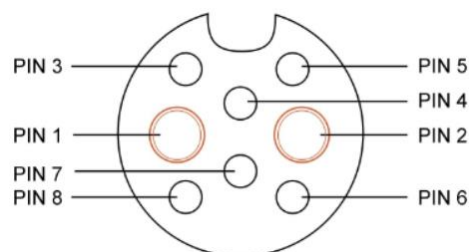


Nota. Tomado de (Ufactory, 2022)

El efector final del brazo robótico UFactory 850 colocado en la parte frontal de la muñeca del prototipo, se emplea para la instalación de herramientas especializadas como pinzas, dispositivos de vacío, entre otros, permitiendo la ejecución directa de diversas tareas. Por ello los diferentes dispositivos creados para la conexión en el efector final presentan los pines de manera estandarizada, por tal motivo se puede conectar cualquier elemento en el extremo operativo del mismo fabricante. A continuación, se va a presentar la distribución de pines del efector final en la Figura 27

Figura 27

Pines del efector final del robot UFactory 850



Nota. Tomado de (Ufactory, 2022)

Tabla 5

Pines y su función del efector final del robot UFactory 850

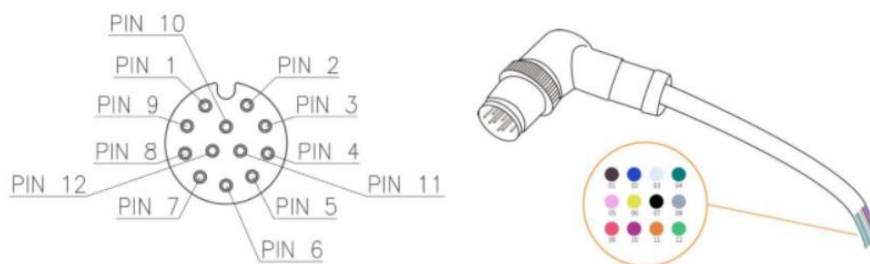
CONECTOR INDUSTRIAL DE 8 PINES

Secuencia de Cables del Conector	Definición Funcional
1	48V
2	GND
3	RS485 A brazo verde
4	Trenza de escudo
5	RS485-B Verde, Brazo Blanco
6	RS485-B Azul, Herramienta Blanco
7	Trenza de escudo
8	RS485-A Herramienta azul

Para las entradas y salidas del efector final en el lado de la herramienta del brazo robótico existe un conector industrial hembra de 12 pines. Este conector proporciona señales de alimentación y control para las pinzas y los sensores utilizados en un brazo robótico concreto ver Figura 28

Figura 28

Pines del conector de entradas y salidas.



Nota. Tomado de (Ufactory, 2022)

Existen 12 pines dentro del cable con diferentes colores, cada color representa diferentes funciones, En la tabla presentada a continuación se indica la funcionalidad de cada pin.

Tabla 6

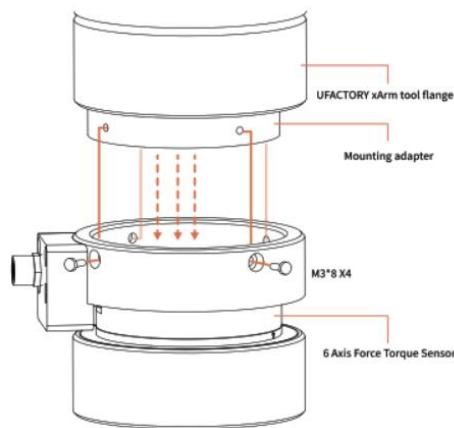
Funcionalidad de pines.

Secuencia de Pines	Color	Señal
1	Café	+24 V (Alimentación)
2	Azul	+24 V (Alimentación)
3	Blanco	0 V (GND)
4	Verde	0 V (GND)
5	Rosado	Pin 485 (A)
6	Amarillo	Pin 485 (B)
7	Negro	Salida de herramienta 0 (TQ0)
8	Gris	Salida de herramienta 1 (TQ1)
9	Rojo	Entrada de herramienta 0 (TI0)
10	Morado	Entrada de herramienta 1 (TI1)
11	Tomate	Entrada analógica 0 (AI0)
12	Verde Claro	Entrada analógica 1 (AI1)

El sensor de fuerza y torque de 6 ejes xArm está diseñado para la adquisición de datos de fuerza y par en xArm, puede medir simultáneamente la fuerza y el par en un espacio tridimensional. Se instala en la brida de la herramienta xArm, se utiliza un cable flexible para la alimentación y las comunicaciones, ver Figura 29

Figura 29

Pines del conector de entradas y salidas.



Nota. Tomado de (Ufactory, 2022)

Tabla 7

Especificaciones de fuerza y torque axial.

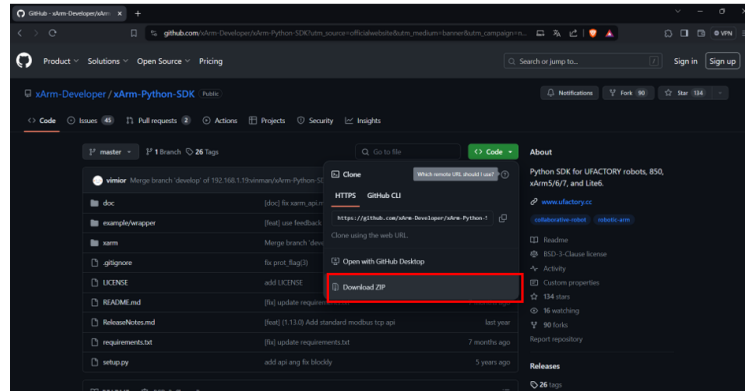
	F_x, F_y	F_z	T_x, T_y, T_z
Capacidad de Carga	150N	200N	4Nm
Resolución	100mN	150mN	5mNm
Histéresis	2.5%FS	1%FS	1%FS
Diafonía	3%FS	3%FS	3%FS
Capacidad de Sobrecarga	150%	150% (F_z+) 300% (F_z-)	150%
Peso			595g

Instalación del software e integración de los SDK Python para el robot Ufactory 850

La Figura 30, muestra la página web de Ufactory donde se encuentran los SDK de Python (Software Development Kit). Los SDK son conjuntos de herramientas que permiten la creación de una aplicación enfocada al control de posicionamiento del

Figura 31

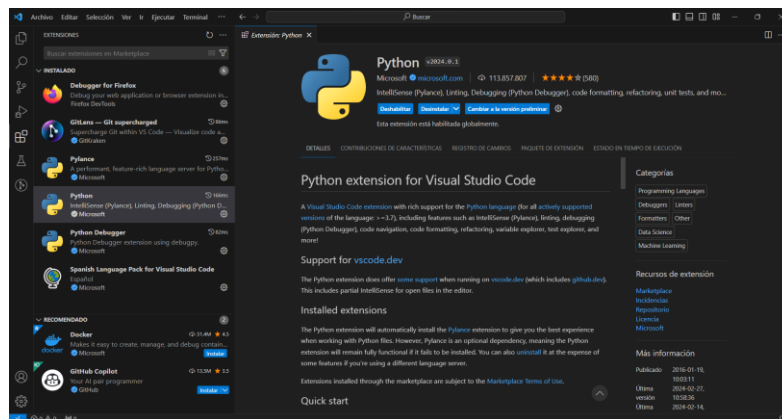
Descarga del archivo ZIP que contiene todos los SDK de Python para el robot Ufactory 850.



En la Figura 32 se observa la carpeta “xArm-Python-SDK-master” para el robot UFactory 850, la cual contiene una serie de archivos y subdirectorios diseñados para facilitar la programación y el control del robot. En su interior, se encuentra módulos y scripts que permiten la comunicación con el robot, Además, la carpeta puede incluir documentación detallada, ejemplos de código, archivos de configuración y recursos adicionales para ayudar a familiarizarse y aprovechar al máximo las capacidades del robot Ufactory 850 mediante el lenguaje de programación de Python.

Figura 33

Integración de Python en el software Visual Studio Code



Para ejecutar y correr programas de Python en el entorno de Visual Studio Code, se recomienda utilizar Git Bash como una interfaz de línea de comandos en sistemas Windows. Git Bash proporciona un entorno similar a Unix en Windows, lo que facilita la ejecución de comandos y scripts de Python de manera fluida. Una vez instalado, se puede abrir Git Bash y utilizarlo como su terminal de línea de comandos para ejecutar y correr programas de Python, así como para gestionar sus proyectos de código fuente utilizando Git para el control de versiones, esto ofrece una solución integral para el desarrollo de software en Python en sistemas Windows como se observa en la Figura 34

Figura 34

Integración de Git-bash en el software Visual Studio Code

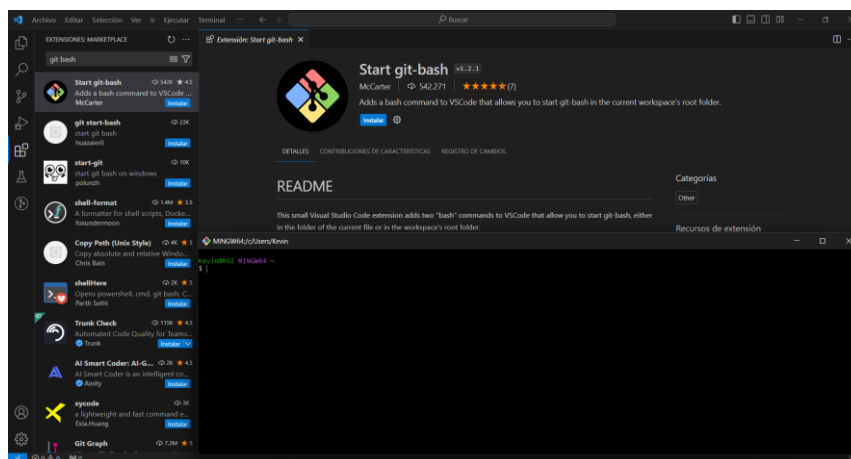
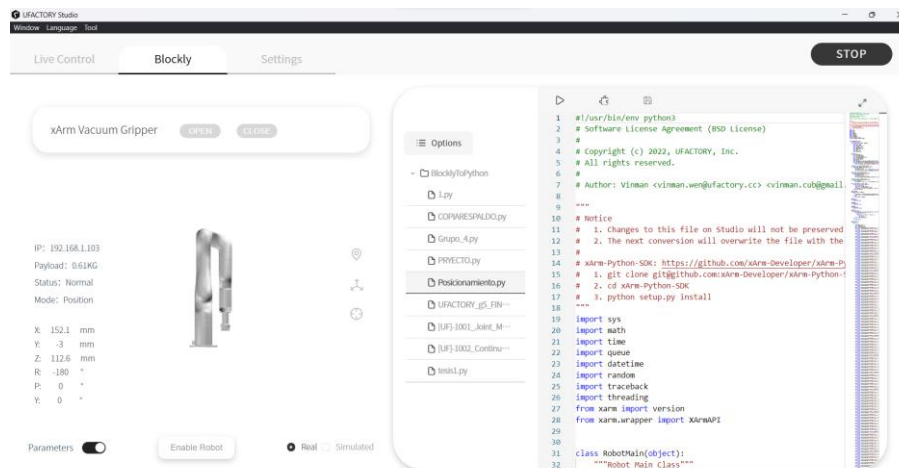


Figura 36

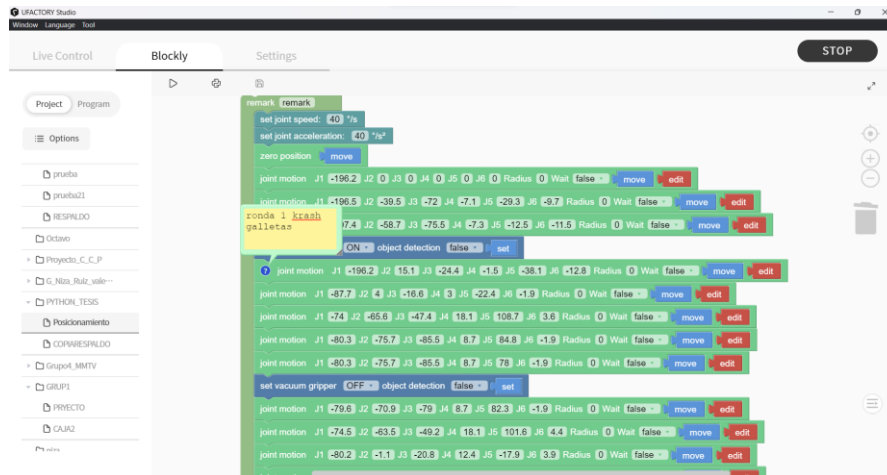
Software Ufactory Studio con funciones de Python.



El software Ufactory Studio ofrece una programación en bloques en donde se puede diseñar y controlar los movimientos y acciones del robot Ufactory 850 mediante la combinación de funciones predefinidas. Estos bloques representan diferentes operaciones y comportamientos del robot, como movimientos de los ejes, acciones de agarre, interacciones con sensores y más como se observa en la Figura 37. Al arrastrar y soltar estos bloques en un lienzo de programación, se puede construir secuencias de acciones paso a paso de manera visual y lógica. Esta metodología de programación en bloques elimina la necesidad de escribir código manualmente, lo que la convierte en una opción ideal tanto para principiantes como para usuarios avanzados que deseen automatizar tareas repetitivas.

Figura 37

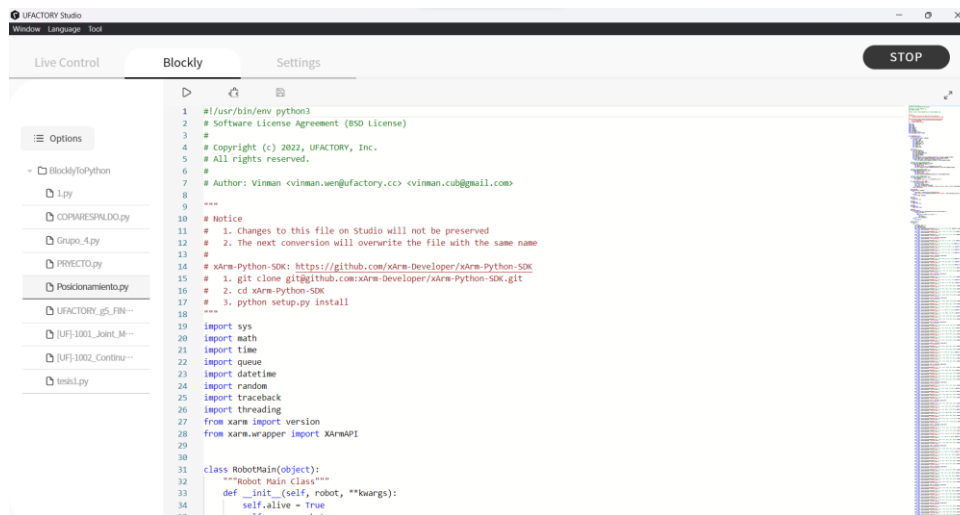
Programación en bloques en el software Ufactory Studio



Además, el software Ufactory Studio proporciona una interfaz integrada para programar el robot Ufactory 850 utilizando Python. Esta característica permite tener las capacidades del lenguaje de programación Python directamente dentro del entorno de Ufactory Studio como se observa en la Figura 38, además, al estar integrado en el mismo entorno se puede alternar fácilmente entre la programación en bloques y la programación en Python según las preferencias y requisitos del proyecto, sin embargo, también se puede ocupar otros softwares de programación en Python para poder ejecutar el programa.

Figura 38

Programación en Python en el software Ufactory Studio



```

1 #!/usr/bin/env python3
2 # Software License Agreement (BSD License)
3 #
4 # Copyright (c) 2022, UFACTORY, Inc.
5 # All rights reserved.
6 #
7 # Author: Vinman <vinman.wen@ufactory.cc> <vinman.cub@gmail.com>
8
9
10 # Notice
11 # 1. Changes to this file on Studio will not be preserved
12 # 2. The next conversion will overwrite the file with the same name
13 #
14 # xArm-Python-SDK: https://github.com/xArm-Developer/xArm-Python-SDK
15 # 1. git clone git@github.com:xArm-Developer/xArm-Python-SDK.git
16 # 2. cd xArm-Python-SDK
17 # 3. python setup.py install
18
19 import sys
20 import math
21 import time
22 import queue
23 import datetime
24 import random
25 import traceback
26 import threading
27 from xarm import version
28 from xarm.wrapper import XArmAPI
29
30
31 class RobotMain(object):
32     """Robot Main Class"""
33     def __init__(self, robot, **kwargs):
34         self.alive = True
35         self.robot = robot
36

```

A Continuación, se presenta un ejemplo de la comparación de la programación de bloques a Python tal y como se observa en la Figura 39, teniendo las siguientes instrucciones en el programa de bloques y su función.

[Set joint speed() %s %s]: Establece la velocidad del movimiento de la articulación en %s.

[Set joint acceleration() %s²]: Establece la aceleración del movimiento de la articulación en %s².

[move joint J1() J2() J3 () J4() J5() J6() J7() ,Radius()]: Establecer cada articulación ángulo para el movimiento de la articulación, la unidad es °.

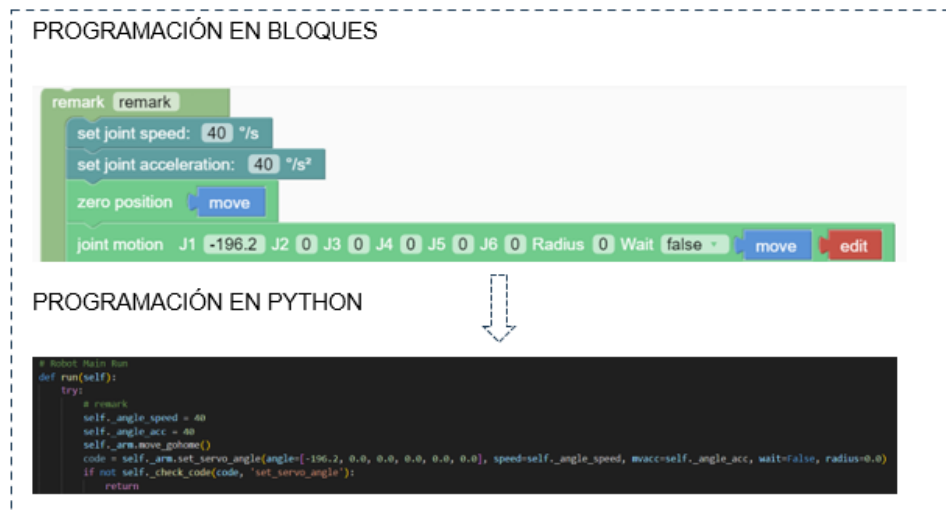
[Wait (true / false)]: indica si se debe esperar a la ejecución de este comando antes de enviar el siguiente comando.

[Move]: El brazo robótico se moverá a la posición actual.

[Edit]: Abre la interfaz de control en vivo y ajusta las coordenadas del punto actual.

Figura 39

Conversión de programación en bloques a programación en Python



Una vez cumplido, el sistema estará listo para ejecutar las aplicaciones y programas destinados a interactuar con el robot UFactory 850. Una vez que se han configurado todas las dependencias y se ha establecido la conexión adecuada con el robot como se muestra en la Figura 40, ya se puede comenzar a desarrollar y probar aplicaciones utilizando las herramientas y recursos proporcionados por los SDK. con todo configurado y listo para su uso.

Figura 40

Programación en Python en el software Visual Studio Code

```

1 #!/usr/bin/env python
2 # Software License Agreement (BSD License)
3 #
4 # Copyright (c) 2022, UFACTORY, Inc.
5 # All rights reserved.
6 #
7 # Author: Vivanan cvivanan@ufactory.cc cvivanan.cuh@gmail.com
8
9
10 # Notice
11 # 1. Changes to this file on Studio will not be preserved
12 # 2. The most conversion will overwrite the file with the same name
13 #
14 # xarm-python-sdk: https://github.com/xarm-developer/xarm-python-sdk
15 # 1. git clone git@github.com:xarm-developer/xarm-python-sdk.git
16 # 2. cd xarm-python-sdk
17 # 3. python setup.py install
18
19
20 import sys
21 import math
22 import time
23 import queue
24 import random
25 import traceback
26 import threading
27 import os
28 import math
29 sys.path.append(os.path.join(os.path.dirname(__file__), '../..'))
30 from xarm import version
31 from xarm.wrapper import XArmAPI
32

```


Capítulo IV

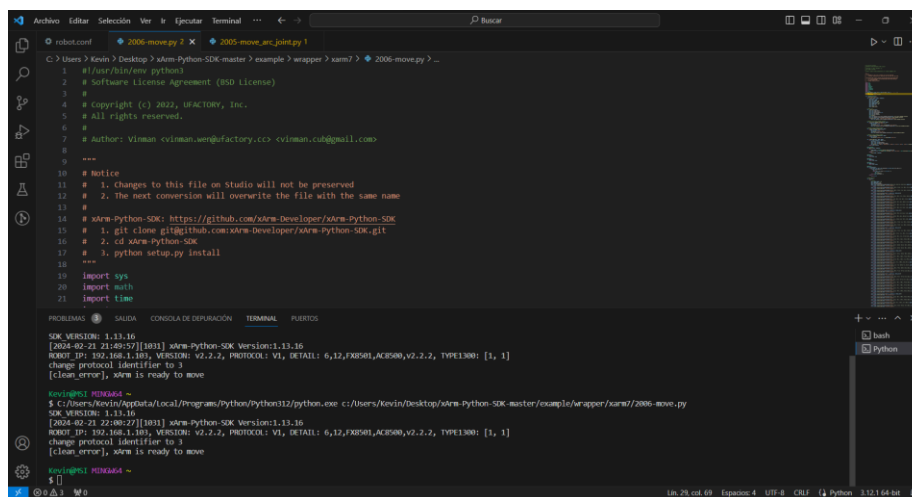
Pruebas y Resultados

En el ámbito de la robótica de servicio, la búsqueda constante de métodos de control para mejorar el rendimiento y la versatilidad de los sistemas automatizados que brindan beneficios a los usuarios para realizar diversas tareas, con el propósito de optimizar el funcionamiento y el tiempo de respuesta. Motivo por el cual, este trabajo presenta la implementación de un control de posición específico para el Brazo Robótico UFactory 850, utilizando el Software Visual Studio Code de Python, el mismo que permite explorar las capacidades y limitaciones de la plataforma desarrollada en la configuración y control de este brazo robótico. A continuación, se presentan las pruebas realizadas y los resultados obtenidos, destacando las eficiencias y posibles áreas de mejora en la implementación del control de posición para este sistema robótico.

En la Figura 41 se muestra la ejecución del programa en el Software Visual Code Studio, mismo que cumple con la tarea previamente programada y hace que el brazo robótico cumpla con la tarea de ordenar las galletas en el casillero del armario que corresponda.

Figura 41

Programa en Visual Code Studio para ejecutar la tarea de posicionamiento



```

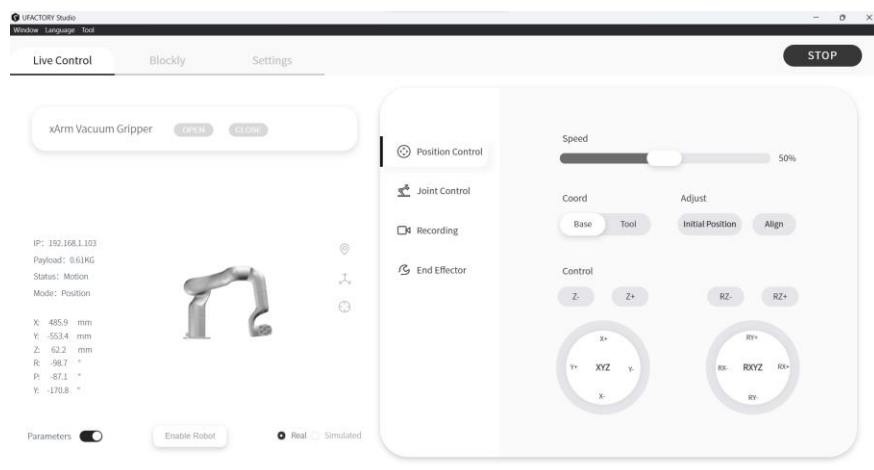
1 # robot.conf
2 # 2006-move.py
3 # 2006-move.py
4 # Software License Agreement (BSD License)
5 #
6 # Copyright (c) 2022, UFACTORY, Inc.
7 # All rights reserved.
8 #
9 # Author: Vinman (vinman.uw@ufactory.cc) (vinman.cub@gmail.com)
10 #
11 #
12 # 1. Changes to this file on Studio will not be preserved
13 # 2. The next conversion will overwrite the file with the same name
14 #
15 # xArm-Python-SDK: https://github.com/xArm-Developer/xArm-Python-SDK
16 # 1. git clone git@github.com:xArm-Developer/xArm-Python-SDK.git
17 # 2. cd xArm-Python-SDK
18 # 3. python setup.py install
19 #
20 #
21 #
22 #
23 #
24 #
25 #
26 #
27 #
28 #
29 #
30 #
31 #
32 #
33 #
34 #
35 #
36 #
37 #
38 #
39 #
40 #
41 #
42 #
43 #
44 #
45 #
46 #
47 #
48 #
49 #
50 #
51 #
52 #
53 #
54 #
55 #
56 #
57 #
58 #
59 #
60 #
61 #
62 #
63 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #
101 #
102 #
103 #
104 #
105 #
106 #
107 #
108 #
109 #
110 #
111 #
112 #
113 #
114 #
115 #
116 #
117 #
118 #
119 #
120 #
121 #
122 #
123 #
124 #
125 #
126 #
127 #
128 #
129 #
130 #
131 #
132 #
133 #
134 #
135 #
136 #
137 #
138 #
139 #
140 #
141 #
142 #
143 #
144 #
145 #
146 #
147 #
148 #
149 #
150 #
151 #
152 #
153 #
154 #
155 #
156 #
157 #
158 #
159 #
160 #
161 #
162 #
163 #
164 #
165 #
166 #
167 #
168 #
169 #
170 #
171 #
172 #
173 #
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #
221 #
222 #
223 #
224 #
225 #
226 #
227 #
228 #
229 #
230 #
231 #
232 #
233 #
234 #
235 #
236 #
237 #
238 #
239 #
240 #
241 #
242 #
243 #
244 #
245 #
246 #
247 #
248 #
249 #
250 #
251 #
252 #
253 #
254 #
255 #
256 #
257 #
258 #
259 #
260 #
261 #
262 #
263 #
264 #
265 #
266 #
267 #
268 #
269 #
270 #
271 #
272 #
273 #
274 #
275 #
276 #
277 #
278 #
279 #
280 #
281 #
282 #
283 #
284 #
285 #
286 #
287 #
288 #
289 #
290 #
291 #
292 #
293 #
294 #
295 #
296 #
297 #
298 #
299 #
300 #
301 #
302 #
303 #
304 #
305 #
306 #
307 #
308 #
309 #
310 #
311 #
312 #
313 #
314 #
315 #
316 #
317 #
318 #
319 #
320 #
321 #
322 #
323 #
324 #
325 #
326 #
327 #
328 #
329 #
330 #
331 #
332 #
333 #
334 #
335 #
336 #
337 #
338 #
339 #
340 #
341 #
342 #
343 #
344 #
345 #
346 #
347 #
348 #
349 #
350 #
351 #
352 #
353 #
354 #
355 #
356 #
357 #
358 #
359 #
360 #
361 #
362 #
363 #
364 #
365 #
366 #
367 #
368 #
369 #
370 #
371 #
372 #
373 #
374 #
375 #
376 #
377 #
378 #
379 #
380 #
381 #
382 #
383 #
384 #
385 #
386 #
387 #
388 #
389 #
390 #
391 #
392 #
393 #
394 #
395 #
396 #
397 #
398 #
399 #
400 #
401 #
402 #
403 #
404 #
405 #
406 #
407 #
408 #
409 #
410 #
411 #
412 #
413 #
414 #
415 #
416 #
417 #
418 #
419 #
420 #
421 #
422 #
423 #
424 #
425 #
426 #
427 #
428 #
429 #
430 #
431 #
432 #
433 #
434 #
435 #
436 #
437 #
438 #
439 #
440 #
441 #
442 #
443 #
444 #
445 #
446 #
447 #
448 #
449 #
450 #
451 #
452 #
453 #
454 #
455 #
456 #
457 #
458 #
459 #
460 #
461 #
462 #
463 #
464 #
465 #
466 #
467 #
468 #
469 #
470 #
471 #
472 #
473 #
474 #
475 #
476 #
477 #
478 #
479 #
480 #
481 #
482 #
483 #
484 #
485 #
486 #
487 #
488 #
489 #
490 #
491 #
492 #
493 #
494 #
495 #
496 #
497 #
498 #
499 #
500 #
501 #
502 #
503 #
504 #
505 #
506 #
507 #
508 #
509 #
510 #
511 #
512 #
513 #
514 #
515 #
516 #
517 #
518 #
519 #
520 #
521 #
522 #
523 #
524 #
525 #
526 #
527 #
528 #
529 #
530 #
531 #
532 #
533 #
534 #
535 #
536 #
537 #
538 #
539 #
540 #
541 #
542 #
543 #
544 #
545 #
546 #
547 #
548 #
549 #
550 #
551 #
552 #
553 #
554 #
555 #
556 #
557 #
558 #
559 #
560 #
561 #
562 #
563 #
564 #
565 #
566 #
567 #
568 #
569 #
570 #
571 #
572 #
573 #
574 #
575 #
576 #
577 #
578 #
579 #
580 #
581 #
582 #
583 #
584 #
585 #
586 #
587 #
588 #
589 #
590 #
591 #
592 #
593 #
594 #
595 #
596 #
597 #
598 #
599 #
600 #
601 #
602 #
603 #
604 #
605 #
606 #
607 #
608 #
609 #
610 #
611 #
612 #
613 #
614 #
615 #
616 #
617 #
618 #
619 #
620 #
621 #
622 #
623 #
624 #
625 #
626 #
627 #
628 #
629 #
630 #
631 #
632 #
633 #
634 #
635 #
636 #
637 #
638 #
639 #
640 #
641 #
642 #
643 #
644 #
645 #
646 #
647 #
648 #
649 #
650 #
651 #
652 #
653 #
654 #
655 #
656 #
657 #
658 #
659 #
660 #
661 #
662 #
663 #
664 #
665 #
666 #
667 #
668 #
669 #
670 #
671 #
672 #
673 #
674 #
675 #
676 #
677 #
678 #
679 #
680 #
681 #
682 #
683 #
684 #
685 #
686 #
687 #
688 #
689 #
690 #
691 #
692 #
693 #
694 #
695 #
696 #
697 #
698 #
699 #
700 #
701 #
702 #
703 #
704 #
705 #
706 #
707 #
708 #
709 #
710 #
711 #
712 #
713 #
714 #
715 #
716 #
717 #
718 #
719 #
720 #
721 #
722 #
723 #
724 #
725 #
726 #
727 #
728 #
729 #
730 #
731 #
732 #
733 #
734 #
735 #
736 #
737 #
738 #
739 #
740 #
741 #
742 #
743 #
744 #
745 #
746 #
747 #
748 #
749 #
750 #
751 #
752 #
753 #
754 #
755 #
756 #
757 #
758 #
759 #
760 #
761 #
762 #
763 #
764 #
765 #
766 #
767 #
768 #
769 #
770 #
771 #
772 #
773 #
774 #
775 #
776 #
777 #
778 #
779 #
780 #
781 #
782 #
783 #
784 #
785 #
786 #
787 #
788 #
789 #
790 #
791 #
792 #
793 #
794 #
795 #
796 #
797 #
798 #
799 #
800 #
801 #
802 #
803 #
804 #
805 #
806 #
807 #
808 #
809 #
810 #
811 #
812 #
813 #
814 #
815 #
816 #
817 #
818 #
819 #
820 #
821 #
822 #
823 #
824 #
825 #
826 #
827 #
828 #
829 #
830 #
831 #
832 #
833 #
834 #
835 #
836 #
837 #
838 #
839 #
840 #
841 #
842 #
843 #
844 #
845 #
846 #
847 #
848 #
849 #
850 #
851 #
852 #
853 #
854 #
855 #
856 #
857 #
858 #
859 #
860 #
861 #
862 #
863 #
864 #
865 #
866 #
867 #
868 #
869 #
870 #
871 #
872 #
873 #
874 #
875 #
876 #
877 #
878 #
879 #
880 #
881 #
882 #
883 #
884 #
885 #
886 #
887 #
888 #
889 #
890 #
891 #
892 #
893 #
894 #
895 #
896 #
897 #
898 #
899 #
900 #
901 #
902 #
903 #
904 #
905 #
906 #
907 #
908 #
909 #
910 #
911 #
912 #
913 #
914 #
915 #
916 #
917 #
918 #
919 #
920 #
921 #
922 #
923 #
924 #
925 #
926 #
927 #
928 #
929 #
930 #
931 #
932 #
933 #
934 #
935 #
936 #
937 #
938 #
939 #
940 #
941 #
942 #
943 #
944 #
945 #
946 #
947 #
948 #
949 #
950 #
951 #
952 #
953 #
954 #
955 #
956 #
957 #
958 #
959 #
960 #
961 #
962 #
963 #
964 #
965 #
966 #
967 #
968 #
969 #
970 #
971 #
972 #
973 #
974 #
975 #
976 #
977 #
978 #
979 #
980 #
981 #
982 #
983 #
984 #
985 #
986 #
987 #
988 #
989 #
990 #
991 #
992 #
993 #
994 #
995 #
996 #
997 #
998 #
999 #
1000 #

```

Cuando se ejecuta el programa el brazo robótico entra en funcionamiento, por medio de la interfaz de UFACTORY se visualiza en tiempo real el movimiento del robot. En la Figura 42 se muestran las posiciones del prototipo en los distintos instantes de tiempo, es decir, muestra la ejecución de la tarea del brazo robótica, además presenta información sobre las posiciones exactas en x , y y z así como también el movimiento del extremo operativo en pitch, roll y yaw.

Figura 42

Interfaz de movimiento y posición en el software UFactory Studio



A continuación, se presentan imágenes que muestran el funcionamiento correcto de la ejecución de la tarea de posicionamiento donde el robot ubica los cuatro diferentes tipos de galletas en el casillero que corresponde.

Funcionamiento

La tarea empieza ubicando las cajas de galleta en un estante de almacenamiento, para lo cual el robot procede a realizar la acción ubicando caja por caja. La acción considera el inicio de la tarea desde la parte inferior hasta la parte superior de derecha a izquierda. LA tarea considera cuatro tipos de galletas considerando tamaños diferentes. Con la finalidad de visualizar la versatilidad de robot a la hora de coger la galleta y ubicarla donde corresponda. Cada casillero del armario ocupara dos galletas, es decir, la tarea se repetirá dos veces y finalizara el proceso de posicionamiento.

En la Figura 43 se muestra la posición inicial del robot y de las galletas para su previo funcionamiento de posicionamiento de las mismas. La posición en la que se encuentra el brazo robótico también es la misma condición en la que se ubicara el prototipo al finalizar la tarea.

Figura 43

Posición inicial del robot y del producto previo a iniciar la tarea.



Cuando se ejecuta el programa por medio de Python, el robot inicia la tarea y empieza a ubicar galleta por galleta donde corresponda. En la Figura 44 se visualiza que el brazo robótico empieza cogiendo la primera galleta para su posterior ubicación. Para poder pasar al siguiente tipo de galleta, para esto primero debe acabar con los dos primeros productos del mismo tipo, luego de manera secuencial acabar los tres productos del segundo tipo de galleta y así sucesivamente hasta haber completado el primer grupo de galletas de cuatro marcas diferentes.

Figura 44

Inicio de la tarea de posicionamiento con la primera galleta.



En la Figura 45, se muestra como el brazo robótico ubica la primera galleta de la primera marca en el casillero número uno, por la explicación anterior se sabe que este proceso se dara de forma sucesiva de derecha a izquierda hasta finalizar en el ultimo casillero del nivel cuatro.

Figura 45

Ubicación de la primera galleta en el casillero uno de la estantería.



La galleta número dos de la misma marca anterior se ubicara en el casillero dos del nivel uno, al lado izquierdo de la primera galleta ubicada anteriormente, tal como se observa en la Figura 46, posteriormente el robot empezara a ubicar el otro tipo de galleta hasta

culminar los 3 casilleros del nivel dos y así llevando la misma lógica hasta acabar ubicando la cuarta marca de galletas en el nivel cuatro.

Figura 46

Ubicación de la segunda galleta en el casillero dos de la estantería.



Finalmente para el primer grupo de galletas, en la Figura 47, se observa que se termina de ubicar la última galleta del nivel cuatro para dar paso a la ubicación del segundo grupo de galletas siguiendo la misma lógica, la idea es tener dos productos en cada casillero de los cuatro niveles del armario.

Figura 47

Ubicación de la última galleta del nivel cuatro del primer grupo.



El segundo grupo de galletas estará ubicado en la misma posición inicial en la que empezó el primer grupo, esto con el fin de no variar los puntos de inicio y solo corregir los puntos finales, es decir, repetir la tarea del primer grupo pero ahora se toma en cuenta que ya existe un producto en el casillero y se necesita ubicar la otra galleta delante de la existente.

En la Figura 48 se muestra el inicio del segundo grupo de galletas, se ha explicado anteriormente que la tarea está en ubicar en cada casillero de los cuatro niveles dos productos iguales, entonces, se repetirá el proceso hasta lograr el objetivo.

Figura 48

Ubicación de la primera galleta en el casillero uno del segundo grupo en la estantería.



Si con el tiempo se desea ubicar otra galleta más en un casillero, se debe mantener la lógica explicada, donde el programador repetirá la tarea manteniendo los puntos iniciales y cambiando los puntos finales de ubicación del nuevo producto para que se ubique uno en delante del otro de forma ordenada

Nuevamente se ha terminado la tarea, pero esta vez del segundo grupo de galletas. En la Figura 49 se visualiza que el brazo robótico ha ubicado en el casillero tres del nivel cuatro la última galleta, posterior a eso el robot se ubicara en la posición inicial.

Figura 49

Finalización del posicionamiento de los dos grupos de galletas.



El brazo robótico ha cumplido de manera correcta con la tarea implementada en el programa del software Visual Studio Code, donde se ha podido ubicar dos productos iguales en cada casillero de cuatro niveles con cuatro marcas distintas de galletas y dando la apertura a que se pueda incrementar otro grupo igual de productos, solo variando los puntos de posición final en el programa.

Capítulo V

Conclusiones y Recomendaciones

Conclusiones

- Los brazos robóticos colaborativos, representan una innovación en el campo de la robótica por su flexibilidad y adaptabilidad que los hacen ideales para una amplia gama de aplicaciones. El brazo robótico Ufactory 850 ofrece la capacidad de realizar tareas tanto en entornos industriales como no industriales con su diseño y su interfaz de programación intuitiva que facilitan la manipulación en diversos procesos.
- Al considerar la morfología del robot Ufactory 850 como grados de libertad, sus articulaciones, el espacio de trabajo, etc. se evidencia la capacidad del robot para realizar tareas de posicionamiento con sus movimientos adaptativos a la tarea que se desea realizar.
- El software UFactory Studio permite realizar la programación de la tarea por medio de bloques donde se toman los puntos iniciales, medios y finales de la trayectoria que debe regirse el brazo para ejecutar la tarea de posicionamiento y el mismo software se encarga de convertir dicha programación en lenguaje de alto nivel para Python y así en el software Visual Code Studio poder compilarlo usando los SDK's del fabricante del robot.
- La programación en Python y el uso de sus bibliotecas ofrecen una ventaja significativa para el robot Ufactory 850. Al utilizar Python se tienen acceso a una amplia gama de bibliotecas y herramientas que pueden integrarse fácilmente con el software del robot. Esto permite la implementación de funcionalidades avanzadas, como el procesamiento de imágenes, el aprendizaje automático y la comunicación con otros dispositivos o realizar algoritmos de control ampliando así las capacidades del robot más allá de las tareas básicas de manipulación.

Recomendaciones

- Es importante que al utilizar los SDK de Python se asegure de llamar todas las bibliotecas necesarias en el programa. Si alguna biblioteca no se importa correctamente, puede generar errores que impidan la ejecución del programa. Para evitar este problema, se debe realizar una revisión exhaustiva del código para asegurarse de que todas las bibliotecas necesarias estén correctamente importadas y disponibles en el entorno de ejecución.
- Es útil documentar claramente las dependencias del proyecto y proporcionar instrucciones claras sobre cómo instalarlas para garantizar que otros desarrolladores puedan configurar el entorno correctamente y evitar problemas de ejecución.
- Es importante tener en consideración las limitaciones del robot UFactory 850 es su susceptibilidad a colisiones, lo que puede resultar en la interrupción del proceso y la necesidad de reiniciar desde el principio. Estas colisiones pueden ocurrir debido a una variedad de razones, como la detección inadecuada de obstáculos, errores en la programación o problemas de precisión en los movimientos del robot.

Bibliografía

- Alvarez, B. (2023). *Control y coordinación de UR3 para robot de servicio en actividades logísticas*. Valencia: UNIVERSITAT POLITÈCNICA DE VALÈNCIA. Obtenido de <https://riunet.upv.es/bitstream/handle/10251/198922/Bazaga%20-%20Control%20y%20coordinacion%20de%20UR3%20para%20robot%20de%20servicio%20en%20actividades%20logisticas.pdf?sequence=2&isAllowed=y>
- Americas, I. d. (2020). *Robotica*.
- Anand Nayyar, Akshi Kumar. (2020). *A Roadmap to Industry 4.0: Smart Production, Sharp Business and sustainable development*. Rusia: Springer. Obtenido de https://link.springer.com/chapter/10.1007/978-3-030-14544-6_9
- Andrius Dzedzickis, Ernestas Šutinys, Urte Samukaite-Bubniene and Vytautas Bucinskas. (13 de Diciembre de 2022). Advanced Applications of Industrial Robotics: New Trends and Possibilities. *MDPI*. Obtenido de <https://www.mdpi.com/journal/applsci>
- Claut, N. (2022). *UnderAutomation*. Obtenido de <https://underautomation.com/es>
- Escobar, J. (2019). *DISEÑO DE SISTEMAS DE CONTROL INDUSTRIAL DE ROBOTS*. Ambato. Obtenido de https://repositorio.uta.edu.ec/bitstream/123456789/29952/1/Tesis_1601id.pdf
- IFR. (2018). Frankfurt.
- IFR. (10 de 2021). *International Federation of Robotics*. Obtenido de Robots in Daily Life: https://ifr.org/downloads/hidden/Information_Paper_Robots_in_Daily_Life_v01.pdf?utm_source=CleverReach&utm_medium=email&utm_campaign=Paper+Download&utm_content=Mailing_12323895
- INESEM. (19 de 1 de 2024). *Herramientas finales en robótica industrial*. Obtenido de <https://www.inesem.es/revistadigital/gestion-integrada/herramientas-finales-en-robotica-industrial/>

- Jazar, R. N. (2021). *Theory of Applied Robotics Kinematics, Dynamics, and Control*. New York: Springer Science. doi:10.1007/978-1-4419-1750-8
- LTD., S. U. (2022). *Manual Ufactory 850*. Obtenido de <https://www.ufactory.cc/ufactory-850/>
- Martín Hernández Ordoñez, M. B. (2015). *Robótica, Análisis, modelado, control e implementación*. , Tamaulipas. doi:<http://dx.doi.org/10.3926/oss.18>
- Matjaž Mihelj, T. B. (2010). *Robotics*. Switzerland: Springer Science. doi:<https://doi.org/10.1007/978-3-319-72911-4>
- Nayely Morales-Ramirez, A. M.-H.-H. (2023). *Implementacion de probabilidades a una ontología para la búsqueda de objetos cotidianosdel hogar por un robot de servicio*. Ciudad de Mexico: Universidad Veracruzana. Obtenido de https://rcs.cic.ipn.mx/2023_152_6/Implementacion%20de%20probabilidades%20a%20una%20ontologia%20para%20la%20busqueda%20de%20objetos%20cotidianos%20del%20hogar.pdf
- Olier Caparroso Iván, A. O. (2021). *Dialnet*. Recuperado el 19 de 11 de 2023, de Una introducción a la robótica industrial: <https://dialnet.unirioja.es/servlet/articulo?codigo=5313913>
- Siciliano, B. (2010). *Robotics, Modelling, Planning and Control*. Londres, Reino Unido: Springer. doi:10.1007/978-1-84628-642-1
- Sino-Inst. (2022). *Socio profesional de Medida y Control*. Obtenido de <https://www.drurylandetheatre.com/es/beginners-guide-to-hart-communicators-hart-protocol/>
- Ufactory, Z. (2022). *Ufactory Manual*. Obtenido de <https://www.ufactory.cc/wp-content/uploads/2023/07/UFactory-850-User-Manual-V2.1.0.pdf>

Anexos